Polynomial eigenvalue solver based on tropically scaled Lagrange linearization*

Marc Van Barel† Françoise Tisseur‡

December 25, 2016

Abstract

We propose a novel approach to solve polynomial eigenvalue problems via linearization. The novelty lies in (a) our choice of linearization, which is constructed using input from tropical algebra and the notion of well-separated tropical roots, (b) an appropriate scaling applied to the linearization and (c) a modified stopping criterion for the QZ iterations that takes advantage of the properties of our scaled linearization. Numerical experiments show that our polynomial eigensolver computes all the finite and well-conditioned eigenvalues to high relative accuracy even when they are very different in magnitude.

keywords: polynomial eigenvalue problem, linearization, tropical scaling, well-separated tropical roots, block companion linearization, Lagrange-type linearization

AMS subject classifications: 65F15, 15A22, 15A80, 15A18, 47J10

1 Introduction

Our aim is to design an algorithm that computes all the finite and well-conditioned eigenvalues of matrix polynomials $P(z) = \sum_{i=0}^{d} z^i P_i$ of degree d to a high *relative* precision even when these eigenvalues are very different in magnitude. To this end, we use a Lagrange linearization of P with interpolation nodes chosen as "well-separated" tropical roots of the associated tropical scalar polynomial $\mathrm{t}_\times p(x) = \max_{0\le i\le d} \|P_i\| x^i$. The tropical roots of $\mathrm{t}_\times p$ are the points at which the maximum is attained for at least two values of i for some x. They can be computed in only $O(d)$ operations and they are known to offer order of magnitude approximations to the moduli of the eigenvalues of P, in particular when the norms of the matrices P_i vary widely [6, 18, 19]. Matrix polynomials with such property occur frequently in applications—see the NLEVP collection of test problems [5]. Their associated tropical scalar polynomials $\mathrm{t}_\times p$ have tropical roots that are quite different in magnitude and, as a result, the eigenvalues of P have large variation in their magnitude.

The standard way of computing eigenvalues of $P(z)$, i.e., the roots of $\det P(z) = 0$, is via linearization of P, that is, by computing the eigenvalues of a larger matrix polynomial $L(z) = A - zB$, which is linear in z and has the same eigenvalues as $P(z)$. Then any eigensolver for generalized eigenproblems can be called to find the eigenvalues of L and thereby that of P [17]. Note that special care needs to be taken with this solution process. Indeed solving the polynomial eigenvalue problem (PEP) by applying a backward stable algorithm (e.g., the QZ algorithm) to a linearization can be backward unstable for the PEP [21]. Also, unless the block structure of the linearization is respected (and it is not by standard techniques), the conditioning of the eigenvalues of the larger linear problem $L(z)$ can be worse than that for the original matrix polynomial $P(z)$,

*Version of December 25, 2016.

†Department of Computer Science, KU Leuven, B-3001 Leuven (Heverlee), Belgium (marc.vanbarel@cs.kuleuven.be). Supported by the Research Council KU Leuven, PF/10/002 (Optimization in Engineering Center (OPTEC)), by the Fund for Scientific Research–Flanders (Belgium), G.0828.14N (Multivariate polynomial and rational interpolation and approximation), and by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office, Belgian Network DYSCO (Dynamical Systems, Control, and Optimization).

‡School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (francoise.tisseur@manchester.ac.uk). Supported by EPSRC grant EP/I005293 and by a Royal Society-Wolfson Research Merit Award.

since the class of admissible perturbations is larger [13, 14]. These issues are likely to occur when P has coefficient matrices whose norms vary widely.

With the aim of addressing these issues, an algorithm proposed by Gaubert and Sharify [9] and recently analyzed in [20], uses the tropical roots τ_j of $\mathrm{t}_\times p$ to scale the (eigenvalue) parameter $z = \tau_j \tilde{z}$ and then computes the eigenvalues of the tropically scaled matrix polynomial $\widetilde{P}(\tilde{z}) := (\mathrm{t}_\times p(\tau_j))^{-1} P(z)$ via the QZ algorithm applied to a companion linearization of $\widetilde{P}(\tilde{z})$. This process is repeated for each of the $t \leq d$ distinct tropical roots τ_j, $j = 1, \ldots, t$. Then, after scaling back the computed eigenvalues, an extraction strategy is applied to retrieve the ds eigenvalues of P, where s is the size of P. Although rigorous backward stability results exist for this algorithm [20], it remains expensive with worse case complexity $\mathcal{O}(d^4 s^3)$.

Instead, we propose to use a single call to the QZ algorithm applied to a tropically scaled Lagrange linearization with well-separated tropical roots as interpolation nodes resulting in an algorithm of complexity $\mathcal{O}(d^3 s^3)$. We show through numerical experiments that if we alter the stopping criterion of the QZ algorithm so as to take into account some of the properties of our chosen linearization then our approach computes all the finite and well-conditioned eigenvalues of P to high relative precision.

In Section 2 we introduce our choice of Lagrange linearization. In Section 3 we give numerical evidence of the claim that for certain types of matrix pencils an adapted version of the QZ algorithm computes the well-conditioned finite eigenvalues with high *relative* precision, even when these eigenvalues are very different in magnitude. Section 4 introduces the concept of well-separated tropical roots and shows that when the interpolation nodes are chosen as well-separated tropical roots, the Lagrange-type linearization of Section 2 can be scaled resulting in a matrix pencil that has the structure described in Section 3. Section 5 discusses the sensitivity and stability of the scaled Lagrange linearization. In Section 6 our new algorithm based on a tropically scaled Lagrange linearization is compared to other eigensolvers based on linearization. For all the problems considered, our new algorithm computes eigenvalues with a small backward error. Section 7 gives the conclusions.

2 Lagrange linearization

Let P be a matrix polynomial of degree d. We only consider *regular* matrix polynomials, i.e., square matrix polynomials whose determinant is not identically equal to zero. The grade of a polynomial is any integer number larger than or equal to the degree of the polynomial. A regular matrix pencil $L(z) = A - zB$ is a linearization of a regular matrix polynomial $P(z)$ if there exist unimodular matrices $E(z)$ and $F(z)$ such that

$$E(z)L(z)F(z) = \begin{bmatrix} P(z) & \\ & I \end{bmatrix}.$$

As defined in [15], $L(z)$ is a strong linearization of $P(z)$ of grade $g \geq \deg P$ if it is a linearization and if the reversed matrix pencil $\mathrm{rev}L(z) = zL(z^{-1}) = zA - B$ is a linearization of the reversed matrix polynomial $\mathrm{rev}P(z) = z^g P(z^{-1})$.

Take d different points $\sigma_i \in \mathbb{C}$, $i = 1, 2, \ldots, d$ with corresponding barycentric weights β_i, i.e., $\beta_i^{-1} = \prod_{j \neq i}(\sigma_i - \sigma_j)$. Let the highest degree coefficient of P be denoted as P_d. A Lagrange pencil very similar to one of the forms given in [2, 7] is the following:

$$L(z) = \left[\begin{array}{c|cccc} P_d & \beta_1 P(\sigma_1) & \beta_2 P(\sigma_2) & \cdots & \beta_d P(\sigma_d) \\ \hline -I_s & (z-\sigma_1)I_s & & & \\ -I_s & & (z-\sigma_2)I_s & & \\ \vdots & & & \ddots & \\ -I_s & & & & (z-\sigma_d)I_s \end{array}\right]. \tag{1}$$

The pencil $L(z)$ in (1) is a strong linearization if we consider P as having grade $d+1$.

Theorem 1. *The regular matrix pencil $L(z)$ as defined in (1) is a strong linearization of the regular matrix polynomial $z^{d+1}0 + P(z)$ of grade $d+1$, where $P(z)$ is of degree d with leading matrix coefficient P_d.*

Proof. Consider the following equality

$$\left[\begin{array}{c|cccc} P_d & \beta_1 P(\sigma_1) & \beta_2 P(\sigma_2) & \cdots & \beta_d P(\sigma_d) \\ \hline -I_s & (z-\sigma_1)I_s & & & \\ -I_s & & (z-\sigma_2)I_s & & \\ \vdots & & & \ddots & \\ -I_s & & & & (z-\sigma_d)I_s \end{array}\right] \left[\begin{array}{c} l(z)I_s \\ \hline l_1(z)I_s \\ l_2(z)I_s \\ \vdots \\ l_d(z)I_s \end{array}\right] = \left[\begin{array}{c} P(z) \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right] \tag{2}$$

with $l(z) = \prod_{i=1}^{d}(z-\sigma_i)$ and $l_i(z) = l(z)/(z-\sigma_i)$. Because the second factor $F_1(z)$ in the left-hand side is of full rank for every value of z, this polynomial block column can be extended into a unimodular matrix $F(z) = [F_1(z), F_2(z)]$. Denoting the top block row of the linearization $L(z)$ by $A_1(z)$ and the remaining rows by $A_2(z)$, we have that

$$\left[\begin{array}{c} A_1(z) \\ A_2(z) \end{array}\right] [F_1(z), F_2(z)] = \left[\begin{array}{cc} P(z) & Y(z) \\ 0 & X(z) \end{array}\right].$$

Because $A_2(z)$ has full rank for every z (as well as $F(z)$), we get that $X(z)$ is unimodular. Hence, defining the unimodular matrix $E(z)$ equal to

$$\left[\begin{array}{cc} I & -Y(z)X(z)^{-1} \\ 0 & X(z)^{-1} \end{array}\right],$$

it follows that

$$E(z)L(z)F(z) = \left[\begin{array}{cc} P(z) & 0 \\ 0 & I_{sd} \end{array}\right].$$

Similarly, one can show that $L(z)$ is a strong linearization of $P(z)$ considering $P(z)$ of grade $d+1$. ■

Theorem 2. *If x and y are right and left eigenvectors, respectively, corresponding to the finite eigenvalue λ of the matrix polynomial $P(z)$, the strong linearization $L(z)$ as defined in (1) has the following vectors v and w as right and left eigenvectors corresponding to the same eigenvalue λ:*

$$\begin{aligned} v &= \left[l(\lambda)I_s \quad l_1(\lambda)I_s \quad \cdots \quad l_d(\lambda)I_s\right]^T x \\ w &= \left[l(\lambda)I_s \quad -\beta_1 P(\sigma_1) l_1(\lambda) \quad \cdots \quad -\beta_d P(\sigma_d) l_d(\lambda)\right]^H y \qquad \text{when } \lambda \neq \sigma_i,\ i = 1, 2, \ldots, d \end{aligned}$$

Proof. Multiplying the left and right hand side of equation (2) to the right by the vector x, it follows that when x is a right eigenvector of $P(z)$, the vector v satisfies $L(\lambda)v = 0$. Because the vector

$$\left[l(\lambda)I_s \quad l_1(\lambda)I_s \quad \cdots \quad l_d(\lambda)I_s\right]^T$$

is of full rank for each value of λ, the vector v is nonzero, i.e., it is a right eigenvector of $L(z)$ corresponding to the eigenvalue λ.

In the same way, when multiplying the linearization $L(\lambda)$ to the left by the vector

$$y^H \left[l(\lambda)I_s \quad -\beta_1 P(\sigma_1) l_1(\lambda) \quad \cdots \quad -\beta_d P(\sigma_d) l_d(\lambda)\right],$$

we get

$$\left[y^H P(\lambda) \quad 0 \quad \cdots \quad 0\right].$$

Hence, if y is a left eigenvector of $P(z)$ corresponding to eigenvalue λ, the vector w, when it is nonzero, is a left eigenvector of $L(z)$ corresponding to the same eigenvalue λ. When λ is different from each of the interpolation nodes σ_i, $i = 1, 2, \ldots, d$, the first block component of w is clearly different from zero. ■

Remark. *The $(d+1)s \times (d+1)s$ Lagrange linearization (1) has the same eigenvalues as $P(z)$ plus an extra n eigenvalues at infinity. Linearizations of size $ds \times ds$ for $s \times s$ matrix polynomials of degree d expressed in the Lagrange basis exist [22, Thm. 4.8]. However, at this point, we were not able to use a scaling/balancing strategy leading to accurate eigenvalues using these $ds \times ds$ linearization. The reason may be due to the fact that with the linearization (1), there is a clear distinction between the top block row defining the coefficients of the matrix polynomial $P(z)$ with respect to the basis as*

it is determined by the remaining block rows (see (2)). Such distinction between the basis on one hand and the coefficients with respect to this basis of a specific matrix polynomial on the other hand, allows the use of the same framework to perform a structured backward error analysis for different choices of bases in [16]. We note that the trivial eigenvalues at infinity, that is, those associated with the first block column of A, can easily be deflated after computing a QR factorization of the first block column or block row of the linearization. Finally we comment that numerically, we do not need the strong linearization property.

3 Accurate eigenvalues for certain types of matrix pencils

Consider an $m \times m$ matrix pencil $A - zB$, where

(i) A has entries of the form $a_{ij} = x + \mathrm{i}y$ with x and y pseudorandomly chosen from the standard normal distribution and $\mathrm{i} = \sqrt{-1}$ so that the entries of A have magnitude of order one (in MATLAB notation, `A = randn(m)+i*randn(m)`),

(ii) B is a block diagonal matrix with first diagonal block possibly equal to zero (implying trivial eigenvalues at infinity $A - zB$) and the other $s \times s$ diagonal blocks being diagonal matrices or dense matrices but in contrast to the magnitude of the entries in A, entries from different diagonal blocks in B can be very different in magnitude. The diagonal blocks except the possibly first zero block can be in any order on the diagonal of matrix B.

The large difference in magnitude in the entries of B leads generically to a large difference in the magnitude of the eigenvalues of $A - zB$. When this difference in magnitude is less than $\epsilon_{\mathrm{mach}}^{-1}$, the QZ algorithm as implemented in LAPACK (and used by MATLAB) computes eigenvalues that have a small relative error (when well-conditioned). However, when this difference is of the order larger than $\epsilon_{\mathrm{mach}}^{-1}$, the LAPACK implementation of the QZ algorithm decides too fast that the corresponding computed eigenvalues are infinite. To address this issue, we modify the LAPACK routine `ZHGEQZ` slightly such that besides the trivial eigenvalues at infinity only finite eigenvalues are generated which can be large when they correspond to exact infinite eigenvalues. More precisely, at two places in the fortran code, we replace the value of `BTOL` by the smallest positive nonzero floating point number avoiding that a specific element of the B matrix is explicitly set to zero, leading to a computed infinite eigenvalue. At the same time, the maximum number of iterations `MAXIT` is increased. To summarize, our algorithm to compute the eigenvalues of the structured matrix pencil $A - zB$ consists of three steps:

- if the first block of matrix B is zero, deflate the trivial infinite eigenvalues based on the first block column;
- reduce the matrix pencil to generalized Hessenberg form;
- compute the generalized Schur form using our modified version of `ZHGEQZ`.

The algorithm then returns the upper triangular pencil $T_A - zT_B$. The finite eigenvalues λ are obtained as

$$\lambda = \frac{\alpha}{\beta},$$

where α and β are the corresponding diagonal elements of T_A and T_B, respectively. The magnitude of α will be of order 10^0 while the magnitude of β determines the magnitude of λ.

We performed several numerical experiments using this strategy and did not find any example where our method does not compute the well-conditioned eigenvalues with high relative precision. We illustrate the behaviour of our adaptation of the QZ algorithm by performing the following experiments.

Numerical example 1: The size s of the diagonal blocks of B is 2, the first diagonal block being the 2×2 zero matrix and the other 16 diagonal blocks being dense matrices with entries generated similarly to the entries of A but scaled with factors $10^{-5}, 10^{-4}, \ldots, 10^{10}$. These blocks are then permuted so that B does not have a graded structure. Such matrix pencils have eigenvalues with order of magnitudes lying between 10^{-10} and 10^{5}. For 100 sample matrices, the left plot in Figure 1 shows the relative errors for the eigenvalues computed by `ZHGEQZ` and ordered from

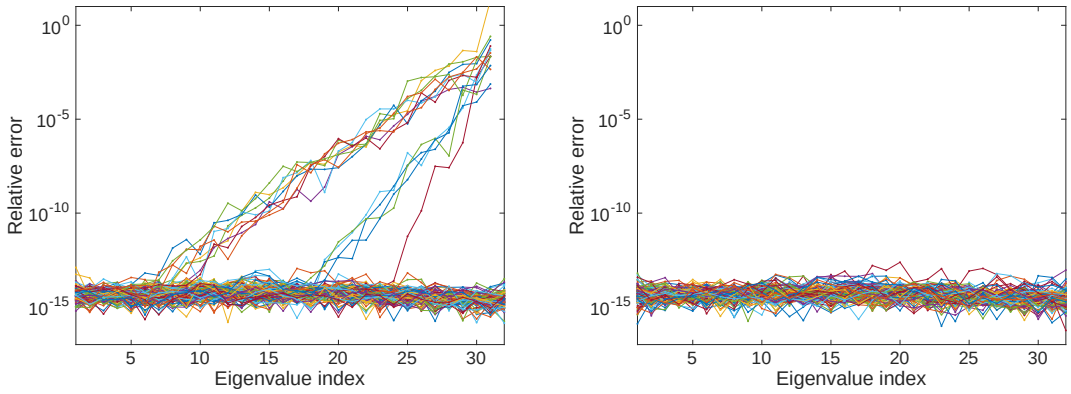

Figure 1: Relative error of computed eigenvalues of structured matrix pencils as defined in Example 1 using the unaltered implementation of `ZHGEQZ` (left) and using our adaptation of the QZ LAPACK implementation (right).

small to large in modulus. For the same set of matrices, the right plot in Figure 1 displays the relative errors of the eigenvalues computed by our modification of `ZHGEQZ`. The "exact" eigenvalues are computed in high precision using the Multiprecision Computing Toolbox for MATLAB from Advanpix [1]. For each pencil in the test set, the corresponding relative errors are joined by straight lines. We see that the unaltered version of `ZHGEQZ` computes accurate eigenvalues for a lot of samples but there are samples such that only the smallest eigenvalues are relatively accurate while the accuracy decreases in the same degree as the magnitude of the eigenvalue increases. One can check that in all these cases at least one of the larger eigenvalues has been approximated by an eigenvalue at infinity.

There is a rule of thumb that says that, to first order in the perturbation, the forward error is bounded above by the condition number times the backward error. The QZ algorithm as implemented in LAPACK is numerically stable in the sense that the computed eigenvalues are the exact eigenvalues of a slightly perturbed pencil $(A + \Delta A) - z(B + \Delta B)$, where the perturbations are such that $\|\Delta A\| \leq p_A(n)\|A\|\epsilon_{\text{mach}}$ and $\|\Delta B\| \leq p_B(n)\|B\|\epsilon_{\text{mach}}$ with $p_A(n), p_B(n)$ low degree polynomials in n and ϵ_{mach} the machine precision. So the left plot in Figure 1 shows that some of the large eigenvalues in magnitude have large condition numbers with respect to unstructured perturbations of the pencils $A - zB$ (we confirmed this by computing the unstructured eigenvalue condition number $\kappa(\lambda)$ in (28)). This behaviour is not observed on the right plot in Figure 1, which suggests that the perturbations generated during the QZ iterations with our modified stopping criterion are not general perturbations but that they are structured perturbations.

Numerical example 2: The data is generated similarly to Example 1 but the factors multiplying the diagonal blocks of B are taken to the power 4, i.e., $10^{-20}, 10^{-16}, \ldots, 10^{40}$. Eigenvalues of such pencils have magnitudes of the order 10^{-40} up to 10^{20}. We take 100 samples and plot the relative errors for the computed eigenvalues ordered from small to large in modulus in Figure 2 only for our modified version of `ZHGEQZ` because the unadapted version struggles to return meaningful eigenvalues in this case because their (unstructured) condition number is too large.

In Section 5 a more detailed explanation for the good behaviour of our adaptation is given in terms of the graded structure of the backward error as well as the graded structure of the left and right eigenvectors. This is not done for the general case but directly for the scaled linearization for the polynomial eigenvalue problem that we describe in the next section. However, similar results are valid for the more general case even when the diagonal blocks of B are not ordered in magnitude as in our previous examples.

4 Interpolation nodes and balancing strategy

We now show how to choose the interpolation nodes for the Lagrange linearization L in (1) together with a specific balancing of the matrices such that the resulting matrix pencil has the structure

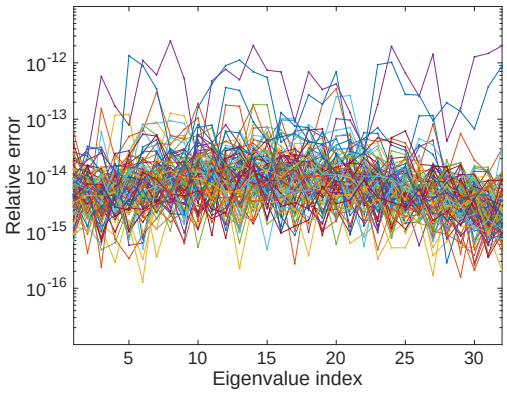

Figure 2: Relative error of computed eigenvalues of structured matrix pencils as defined in Example 2 using our adaptation of the QZ LAPACK implementation

and property of those in Section 3.

4.1 Choice of the points σ_i (part 1)

Consider the Lagrange linearization $L(z) = A - zB$ in (1). Before constructing L, we scale the matrix polynomial $P(z)$ such that the norm of the highest degree coefficient P_d becomes 1. This is equivalent to a scaling of the first block row of A and B. We can scale all block columns (except the first one) by σ_i^{-1}, leading to the following matrix pencil

$$L_1(z) = A_1 - zB_1 = \left[\begin{array}{c|cccc} P_d & \beta_1 P(\sigma_1)/\sigma_1 & \beta_2 P(\sigma_2)/\sigma_2 & \cdots & \beta_d P(\sigma_d)/\sigma_d \\ \hline -I_s & (z/\sigma_1 - 1)I_s & & & \\ -I_s & & (z/\sigma_2 - 1)I_s & & \\ \vdots & & & \ddots & \\ -I_s & & & & (z/\sigma_d - 1)I_s \end{array}\right]. \tag{3}$$

Except for the first block row of A_1, all other nonzero elements of that matrix are equal to -1. The matrix B_1 is a (block-)diagonal matrix with nonzero diagonal blocks $-\sigma_i^{-1} I_s$.

If we can choose the points σ_i such that

$$\frac{\|P(\sigma_i)\||\beta_i|}{|\sigma_i|} = \mathcal{O}(1), \quad i = 1, \ldots, d \tag{4}$$

then all the nonzero blocks of A_1 will have norm of order one, and the matrix pencil $A_1 - zB_1$ will have the properties described in Section 3, which allows the computation of eigenvalues (if well-conditioned) to high relative precision even when they are very different in magnitude. Note also that the backward error analysis of standard linearization processes indicates that to compute eigenvalues with small backward errors, it is important for the nonzero blocks of the linearization to have norm close to one (see [12, 14, 21]).

We show next that a possible choice for the points σ_i, $i = 1, \ldots, d$ is based on a set of well-separated tropical roots $\tilde{\tau}_\ell$, $\ell = 1, 2, \ldots, \tilde{t}$. We introduce this type of tropical roots in the next subsection.

4.2 Well-separated tropical roots

Let us first repeat the concept of tropical roots. Given the norms $\|P_j\|$ of the matrix coefficients P_j of the matrix polynomial $P(z) = \sum_{j=0}^{d} P_j z^j$, we define the tropical (or max-times) scalar polynomial

$$\mathrm{t}_\times p(x) = \max_{0 \le j \le d} \|P_j\| x^j,$$

where x takes nonnegative real values. As mentioned in the introduction, the tropical roots of $\mathrm{t}_\times p$ are points at which the maximum is attained for at least two values of j for some x (see for example [18, 19] for a precise definition of tropical roots together with their multiplicities).

The tropical roots and their corresponding multiplicities m_ℓ can easily be obtained using the Newton polygon of $\mathrm{t}_\times p(x)$ defined as the upper boundary of the convex hull of the set of points $(j, \log(\|P_j\|)$, $j = 0, \ldots, d$. If we denote by

$$k_0 = 0 < \cdots < k_t = d$$

the abscissae of the vertices of the Newton polygon then $\mathrm{t}_\times p(x)$ has t distinct roots given by

$$\tau_\ell = \left(\frac{\|P_{k_{\ell-1}}\|}{\|P_{k_\ell}\|} \right)^{1/m_\ell}, \qquad \ell = 1, \ldots, t, \tag{5}$$

with multiplicities $m_\ell = k_\ell - k_{\ell-1}$, $\ell = 1, \ldots, t$, respectively. Note that $\sum_{\ell=1}^{t} m_\ell = d$, i.e., a tropical scalar polynomial of degree d has d tropical roots counting multiplicities. We have that

$$\begin{aligned} \tau_\ell^{k_{\ell-1}} \|P_{k_{\ell-1}}\| &= \tau_\ell^{k_\ell} \|P_{k_\ell}\| \\ &\geq \tau_\ell^{j} \|P_j\|, \qquad k_{\ell-1} < j < k_\ell \qquad (6) \\ &\geq \tau_{\ell'}^{j} \|P_j\|, \qquad \ell' \neq \ell, \quad k_{\ell'-1} \leq j \leq k_{\ell'}. \qquad (7) \end{aligned}$$

For (4) to hold with σ_i equal to a tropical root, the latter need to be separated in a certain sense. Therefore, we introduce the concept of well-separated tropical roots,

$$\widetilde{\tau}_\ell = \left(\frac{\|P_{\widetilde{k}_{\ell-1}}\|}{\|P_{\widetilde{k}_\ell}\|} \right)^{1/\widetilde{m}_\ell}, \qquad \ell = 1, 2, \ldots, \tilde{t}, \tag{8}$$

$\widetilde{\tau}_\ell$ with corresponding multiplicities $\widetilde{m}_\ell$ and $\tilde{t} \leq t$. They form a subset of the tropical roots but each well-separated tropical root $\widetilde{\tau}_\ell$ has a larger multiplicity $\widetilde{m}_\ell \geq m_\ell$. To determine the indices

$$\widetilde{k}_0 = 0 < \cdots < \widetilde{k}_{\tilde{t}} = d$$

with $\widetilde{k}_\ell \in \{k_0, \ldots, k_t\}$, we add the condition that the numbers $\widetilde{\tau}_\ell$ have to be well-separated in the sense that

$$\widetilde{\tau}_\ell \leq \gamma \widetilde{\tau}_{\ell+1}, \qquad \ell = 1, 2, \ldots, \tilde{t} - 1 \tag{9}$$

for a chosen value of the separation parameter $\gamma < 1$. In this case the inequalities (6)–(7) have to be relaxed as follows:

$$\begin{aligned} \widetilde{\tau}_\ell^{\widetilde{k}_{\ell-1}} \|P_{\widetilde{k}_{\ell-1}}\| &= \widetilde{\tau}_\ell^{\widetilde{k}_\ell} \|P_{\widetilde{k}_\ell}\| \qquad (10) \\ &\geq \rho^{-1} \widetilde{\tau}_\ell^{j} \|P_j\|, \qquad \widetilde{k}_{\ell-1} < j < \widetilde{k}_\ell \qquad (11) \\ &\geq \rho^{-1} \widetilde{\tau}_{\ell'}^{j} \|P_j\|, \qquad \ell' \neq \ell, \quad \widetilde{k}_{\ell'-1} \leq j \leq \widetilde{k}_{\ell'} \qquad (12) \end{aligned}$$

where $\rho \geq 1$ is the relaxation parameter.

Note that (10)–(12) do not determine a unique set of indices $\widetilde{k}_\ell$, $\ell = 1, \ldots, \tilde{t}$. The standard algorithm for computing the tropical roots of $\mathrm{t}_\times p$ based on the convex hull of the points $(j, \|P_j\|)$, $j = 0, 1, \ldots, d$ (see for example [19]) can be extended to compute well-separated tropical roots $\widetilde{\tau}_\ell$. Note that the angles between the different segments of the approximation of the Newton polygon of $\mathrm{t}_\times p(x)$ defined by the indices $\widetilde{k}_\ell$, $\ell = 0, \ldots, \tilde{t}$ are connected to the separation parameter γ while the distance between this approximation of the Newton polygon and the points above it, is linked to the relaxation parameter ρ (see Figure 4).

4.3 Choice of the points σ_i (part 2)

To avoid clutter in the notation, we drop in what follows the tilde in the notation for the well-separated tropical roots. Based on the knowledge of the well-separated tropical roots τ_ℓ and their corresponding multiplicities m_ℓ, the points σ_i are chosen as follows:

$$\{\sigma_i\}_{i=k_{\ell-1}+1}^{k_\ell} = \{m_\ell \text{ roots of unity multiplied by } \tau_\ell\}. \tag{13}$$

Hence, each of the interpolation points σ_i satisfies

$$|\sigma_i| = \tau_\ell, \qquad k_{\ell-1}+1 \leq i \leq k_\ell.$$

Note that in this case

$$\prod_{j=k_{\ell-1},j\neq i}^{k_\ell} (\sigma_i - \sigma_j) = m_\ell \sigma_i^{m_\ell - 1}. \tag{14}$$

4.4 Upper bound for $\frac{\|P(\sigma_i)\|\|\beta_i|}{|\sigma_i|}$

Let us first derive an upper bound for $\|P(\sigma_i)\|$. Suppose σ_i corresponds to the well-separated tropical root τ_ℓ. Hence, we get the following upper bound:

$$\begin{aligned}
\|P(\sigma_i)\| &\leq \sum_{k=0}^{d} |\sigma_i|^k \|P_k\| && (15)\\
&\leq \sum_{k=0}^{k_{\ell-1}-1} |\sigma_i|^k \|P_k\| + \sum_{k=k_{\ell-1}}^{k_\ell} |\sigma_i|^k \|P_k\| + \sum_{k=k_\ell+1}^{d} |\sigma_i|^k \|P_k\| && (16)\\
&\leq \sum_{k=0}^{k_{\ell-1}-1} \tau_\ell^k \|P_k\| + \sum_{k=k_{\ell-1}}^{k_\ell} \tau_\ell^k \|P_k\| + \sum_{k=k_\ell+1}^{d} \tau_\ell^k \|P_k\|. && (17)
\end{aligned}$$

From (10) and (11), an upper bound for each of the terms in this sum is given by

$$\rho \tau_\ell^{k_{\ell-1}} \|P_{k_{\ell-1}}\| = \rho \tau_\ell^{k_\ell} \|P_{k_\ell}\|. \tag{18}$$

Let us now derive an upper bound for $|\beta_i/\sigma_i|$. When $k_{\ell-1} < i \leq k_\ell$, i.e., when σ_i corresponds to the well-separated tropical root τ_ℓ, we can write β_i^{-1} as

$$\begin{aligned}
\beta_i^{-1} &= \prod_{j\neq i} (\sigma_i - \sigma_j) && (19)\\
&= \prod_{j=1}^{k_{\ell-1}} (\sigma_i - \sigma_j) \cdot \prod_{j=k_{\ell-1}+1,j\neq i}^{k_\ell} (\sigma_i - \sigma_j) \cdot \prod_{j=k_\ell+1}^{d} (\sigma_i - \sigma_j) && (20)\\
&= \sigma_i^{k_{\ell-1}} \prod_{j=1}^{k_{\ell-1}} (1 - \frac{\sigma_j}{\sigma_i}) \cdot m_\ell \sigma_i^{m_\ell - 1} \cdot \prod_{j=k_\ell+1}^{d} \sigma_j \prod_{j=k_\ell+1}^{d} (\frac{\sigma_i}{\sigma_j} - 1). && (21)
\end{aligned}$$

Going from (20) to (21), we have pulled out in each factor of the first product σ_i and in each factor of the third product σ_j and we made use of (14).

Using the separation parameter γ and the correspondence between the interpolation points σ_i and the well-separated tropical roots τ_ℓ, gives us the following lower bound for $|\beta_i^{-1}\sigma_i|$

$$|\beta_i^{-1}\sigma_i| \geq m_\ell \tau_\ell^{k_\ell} \Gamma_\ell \prod_{j=k_\ell+1}^{d} |\sigma_j| \tag{22}$$

with

$$\Gamma_\ell = \prod_{j=1,j\neq\ell}^{t} (1 - \gamma^{|\ell-j|})^{m_j}.$$

Hence, an upper bound for the largest term in the sum (17) based on (18) and (22) is given by

$$\rho \tau_\ell^{k_\ell} \|P_{k_\ell}\| \left(m_\ell \tau_\ell^{k_\ell} \Gamma_\ell \prod_{j=k_\ell+1}^{d} |\sigma_j| \right)^{-1}. \tag{23}$$

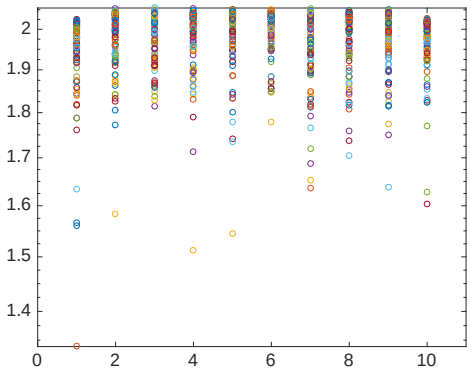

Figure 3: The values $\|P(\sigma_i)\|\|\beta_i|/|\sigma_i|$, $i = 1, 2, \ldots, d$ for well-separated tropical roots

From (8), we know that

$$\frac{\|P_{k_\ell}\|}{\prod_{j=k_\ell+1}^{d} |\sigma_j|} = \|P_d\|$$

so that the upper bound (23) is equal to

$$\frac{\rho}{m_\ell \Gamma_\ell} \|P_d\|. \tag{24}$$

Hence, taking all terms together, we derive the upper bound

$$\frac{\|P(\sigma_i)\|\|\beta_i|}{|\sigma_i|} \leq \frac{\rho(d+1)}{m_\ell \Gamma_\ell} \|P_d\|. \tag{25}$$

Note that when the classical tropical roots are already very well-separated and have multiplicity one, i.e., when $\Gamma_\ell \approx 1$, $\rho = 1$, there will only be 2 terms that are large in the sum and the upper bound can be well approximated by $2\|P_d\|$. This upper bound is obtained in practice as the following numerical experiments illustrate.

4.5 Numerical experiments

Experiment 1: Let us first show that $\|P(\sigma_i)\|\|\beta_i|/|\sigma_i|$ is of order one and bounded above by a number slightly larger than 2 for very well-separated tropical roots having multiplicity one. For that, we take well-separated tropical roots $\tau_\ell = 10^{2(\ell-1)}$, $\ell = 1, 2, \ldots, 10$ having multiplicity $m_\ell = 1$. The coefficient matrices P_i of the matrix polynomial P of degree d are constructed as $\theta_i U_i$, where the U_i are random unitary matrices and the scalars θ_i are chosen such that $\mathrm{t}_\times p(x) = \max_{0 \leq i \leq d} \|P_i\| x^i$ has tropical roots τ_ℓ, $\ell = 1, \ldots, 10$. The highest degree coefficient is scaled such that its norm is equal to one. Figure 3 shows the values $\|P(\sigma_i)\|\|\beta_i|/|\sigma_i|$ for 100 samples for $P(z)$ and confirm the analysis in Section 4.4.
Experiment 2: In this case, we choose the tropical roots not well-separated, e.g., $\tau_\ell = (1.5)^{\ell-1}$, $\ell = 1, 2, \ldots, 20$. Note that $1.5^{19} \approx 2.2 \cdot 10^3$. Hence, the 20 tropical roots are relatively near to each other. We take 100 samples of $P(z)$ constructed as in Experiment 1. The left plot in Figure 5 shows that with the choice $\sigma_i = \tau_i$, $i = 1, \ldots, \ell$, $\|P(\sigma_i)\|\|\beta_i|/|\sigma_i|$ can have large magnitude. The right plot in Figure 5 is obtained by using well-separated tropical roots constructed from P with a separation parameter $\gamma = 5^{-1}$. Note the significant improvement in the boundedness of the magnitude of $\|P(\sigma_i)\|\|\beta_i|/|\sigma_i|$.

The Newton polygon associated with $\mathrm{t}_\times p(x) = \max_{0 \leq j \leq d} \|P_j\| x^j$ for one sample of $P(z)$ is plotted in Figure 4 together with its approximation defining the well-separated roots. The latter are given in Table 1.

4.6 Summary of the balancing strategy

The scaling and balancing strategy that is used in the numerical experiments is summarized as follows:

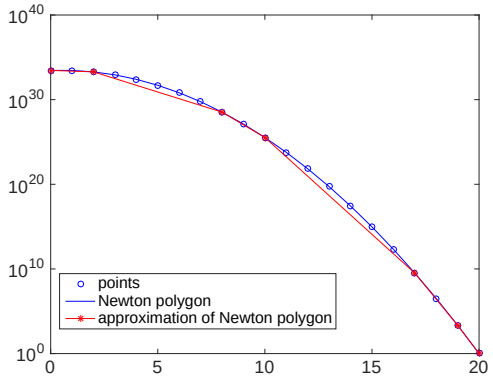

Figure 4: The Newton polygon and its approximation from which the well-separated tropical roots are derived for Example 2 of the numerical experiments

Table 1: Well-separated tropical roots $\widetilde{\tau}_\ell$ and their multiplicities $\widetilde{m}_\ell$ for a sample matrix polynomial from Example 2.

$\widetilde{\tau}_\ell$	1.2e+0	6.2e+0	3.1e+1	1.9e+2	1.2e+3	2.2e3
$\widetilde{m}_\ell$	2	6	2	7	2	1

- Compute well-separated tropical roots τ_ℓ (for a given separation parameter γ) and the corresponding interpolation points σ_i using (13) so that

$$\frac{\|P(\sigma_i)\||\beta_i|}{|\sigma_i|} \leq \frac{\rho(d+1)}{m_\ell \Gamma_\ell}\|P_d\| = \mathcal{O}(1)\|P_d\|$$

  with a value of the relaxation parameter not much larger than one, i.e., $\rho \approx 1$.

- Scale the matrix polynomial $P(z) \leftarrow \frac{P(z)}{\|P_d\|}$ such that the norm of the highest degree coefficient P_d becomes equal to one.

- Construct the Lagrange-type linearization $A - zB$ (3) with the top block row of A equal to

$$\begin{bmatrix} P_d & \dfrac{\beta_1 P(\sigma_1)}{\sigma_1} & \dfrac{\beta_2 P(\sigma_2)}{\sigma_2} & \cdots & \dfrac{\beta_d P(\sigma_d)}{\sigma_d} \end{bmatrix}$$

- Apply the modified QZ algorithm excluding nontrivial roots at infinity.

5 Sensitivity of the scaled Lagrange linearization

The numerical experiments in Section 3 suggest that the backward error of the modified QZ iterations when applied to a matrix pencil of the form described in Section 4.6 is structured. Let us explain this in more detail. Suppose we have a finite eigenvalue λ with corresponding right and left eigenvectors v and w, respectively, for the matrix pencil $zB - A$, i.e.,

$$(\lambda B - A)v = 0, \qquad w^*(\lambda B - A) = 0.$$

Putting an error ΔA on A and ΔB on B, the eigenvalue λ changes into $\lambda + \Delta\lambda$ while the left and right eigenvectors change into $v + \Delta v$ and $w + \Delta w$. It is easy to show that, up to first order, the relative error on a simple finite and nonzero eigenvalue λ is

$$\frac{|\Delta\lambda|}{|\lambda|} = \frac{|w^*(\lambda\Delta B - \Delta A)v|}{|\lambda w^* B v|}. \tag{26}$$

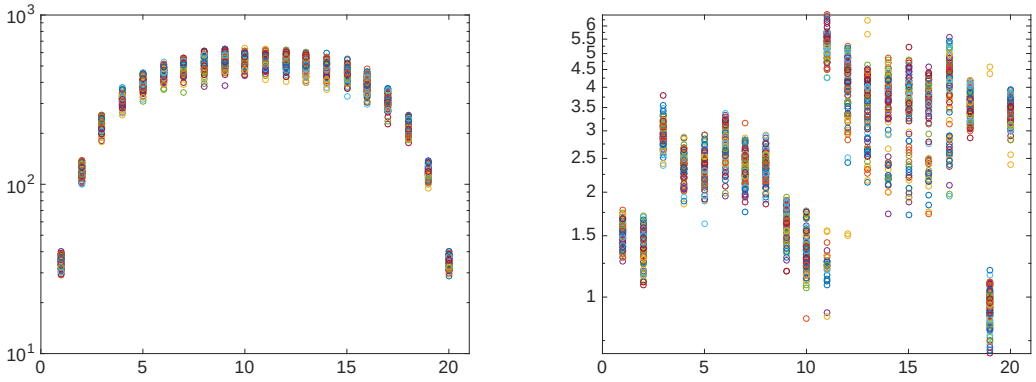

Figure 5: The values $\|P(\sigma_i)\||\beta_i|/|\sigma_i|$, $i = 1, 2, \ldots, d$ for not well-separated (left) and well-separated tropical roots (right)

Taking into account a normwise upper bound for the errors ΔA and ΔB, i.e., $\|\Delta A\| \leq \epsilon\|A\|$ and $\|\Delta B\| \leq \epsilon\|B\|$, leads to an upper bound for the relative error on the eigenvalue

$$\frac{|\Delta\lambda|}{|\lambda|} \leq \kappa(\lambda)\epsilon, \tag{27}$$

where

$$\kappa(\lambda) = \frac{(|\lambda|\|B\| + \|A\|)\|w\|\|v\|}{|\lambda||w^*Bv|} \tag{28}$$

is the eigenvalue condition number of λ as defined in [11]. Now if we let

$$\alpha = w^*Av, \qquad \beta = w^*Bv$$

then $\lambda = \frac{\alpha}{\beta}$. Taking the left and right eigenvector of norm one, the upper bound (27) can be written as

$$\frac{|\Delta\lambda|}{|\lambda|} \leq \left(\frac{\|A\|}{|\alpha|} + \frac{\|B\|}{|\beta|}\right)\epsilon. \tag{29}$$

Because of the assumptions, $|\alpha| \leq \|A\|$ and $|\beta| \leq \|B\|$. To obtain a small upper bound, $|\alpha|$ should be close to $\|A\|$ and $|\beta|$ should be close to $\|B\|$.

Assume now that the pencil is scaled in such a way that $\|A\| = \|B\| = 1$. Note that our balancing strategy leads to a pencil $zB - A$ with $\|A\|$ of order one and $\|B\| = \widetilde{\tau}_1^{-1}$. Without loss of generality, we can assume that the problem is scaled such that the smallest well-separated tropical root $\widetilde{\tau}_1$ is equal to one. Suppose that $\lambda \geq 1$. Because $\lambda = \alpha/\beta$, we can rewrite (29) as

$$\frac{|\Delta\lambda|}{|\lambda|} \leq \left(\frac{\|A\|}{|\alpha|} + \frac{|\lambda|\|B\|}{|\alpha|}\right)\epsilon.$$

Hence, the upper bound will grow when $|\lambda|$ grows. This behaviour is observed on the left plot in Figure 1. The fact that this is not observed on the right plot in Figure 1 when using our adaptation of the QZ LAPACK implementation tells us that the above analysis cannot be used in this case. Instead we can look at formula (26) for the relative error that is valid up to first order. Let us assume that $\|A\|$ and $\|B\|$ are of order one, that $\|v\| = \|w\| = 1$, and that the eigenvalues are of order one or larger. To obtain a relative error of the order of the machine precision ϵ_{mach}, the backward error ΔB (and ΔA) and the left and right eigenvectors should be such that w^*Bv as well as $w^*\Delta Bv$ behave as λ^{-1} and $\lambda^{-1}\epsilon_{\text{mach}}$ respectively, while $w^*\Delta Av$ is of the order of ϵ_{mach}. Currently, we are not able to prove that this is the case for our scaling strategy together with the adapted version of the QZ algorithm. But all the numerical experiments we performed, indicate that this is the case as we now illustrate with an example.

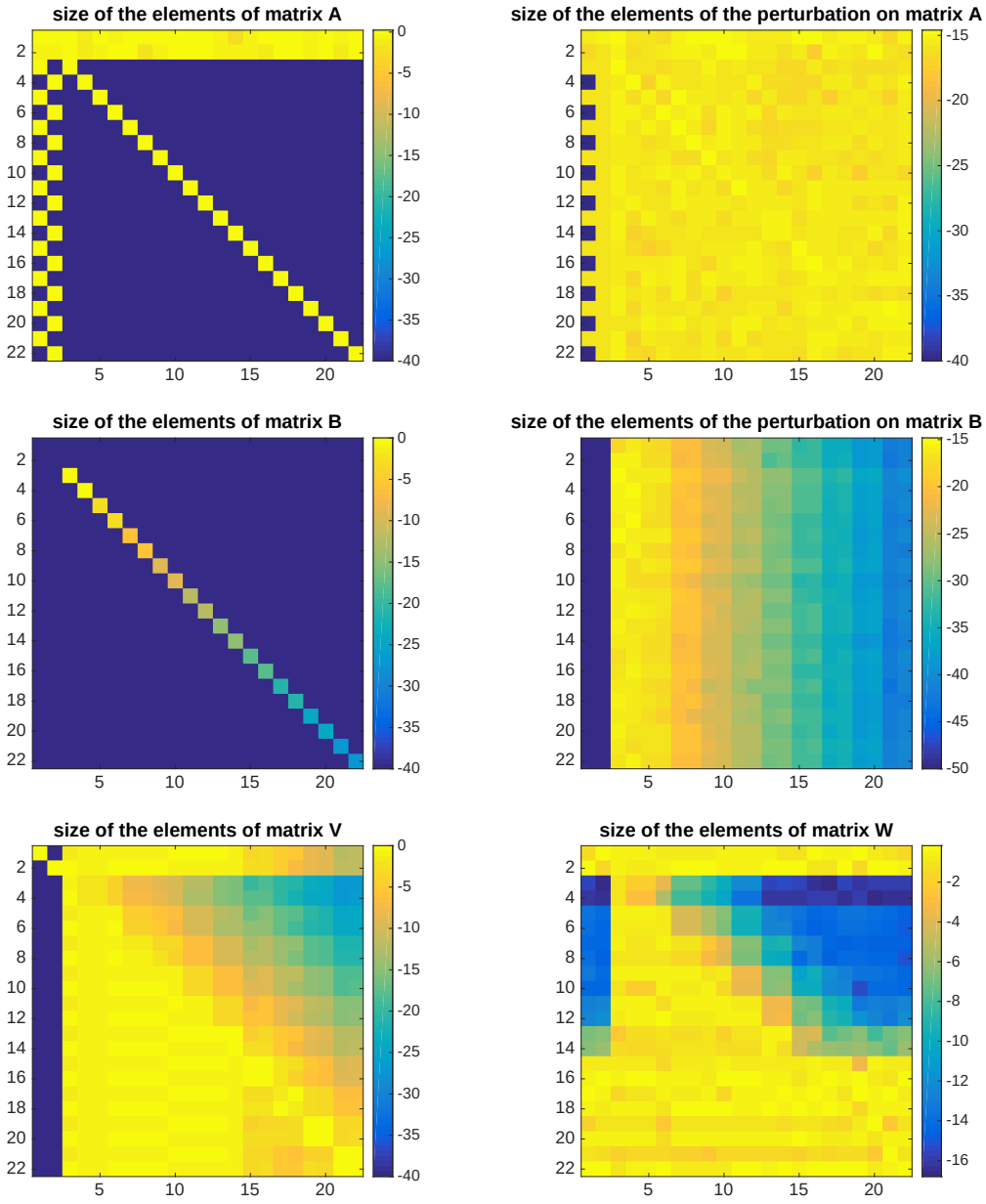

Figure 6: Backward error on matrix A (top) and on matrix B (middle). Graded character of the matrices V and W (bottom).

5.1 Numerical illustration of the graded backward error

Let $P(z)$ be a matrix polynomial of degree $d = 10$ whose coefficients are random 2×2 matrices with entries taken from the standard normal distribution but scaled such that the well-separated tropical roots are $\{\tau_i\} = \{10^0, 10^3, \ldots, 10^{27}\}$ each with multiplicity one. Such matrix polynomial has eigenvalues with magnitude varying between 1 and 10^{27}. Figure 6 (top row) shows the magnitude of the entries of the backward error matrix on A corresponding to computed eigenvalues using our adapted version of the QZ algorithm while the middle row shows the backward error on the matrix B. The backward error matrices are computed using the Multiprecision Computing Toolbox for MATLAB from Advanpix [1]. Note the graded character in the backward error on the matrix B. The graded character of the right and left eigenvector matrices V and W is shown at the bottom of Figure 6. Without this graded character and the graded structure of the backward error on B, it would not be possible to obtain the eigenvalues with high relative accuracy.

6 Numerical experiments

For our numerical experiments we only show the backward error for the computed eigenvalues because the backward error for the corresponding eigenpairs is very similar. Following [21] the backward error $\eta_P(\widetilde{\lambda})$ for a computed finite eigenvalue $\widetilde{\lambda}$ of P can be computed as

$$\eta_P(\widetilde{\lambda}) = \frac{\|P(\tilde{\lambda})^{-1}\|_2^{-1}}{\sum_{i=0}^{d} |\tilde{\lambda}|^i \|P_i\|_2} = \frac{\sigma_{\min}(P(\tilde{\lambda}))}{\sum_{i=0}^{d} |\tilde{\lambda}|^i \|P_i\|_2}. \tag{30}$$

Let $\widetilde{\Lambda}(P; \text{Alg.}j)$ be the set of approximate finite eigenvalues of P computed by some algorithm Alg.j. We denote by

$$\eta_P^{\max}(\text{Alg.}j) = \max\{\eta_P(\widetilde{\lambda}) : \widetilde{\lambda} \in \widetilde{\Lambda}(P; \text{Alg.}j)\} \tag{31}$$

the largest backward error for the eigenvalues of P computed by algorithm Alg.j. We use $\eta_P^{\max}(\text{Alg.}j)$ to compare the backward stability of five different eigensolvers for polynomial eigenvalue problems:

Alg.1 modified LAPACK implementation of the QZ algorithm applied to tropically scaled Lagrange linearization using classical tropical roots (5);

Alg.2 modified LAPACK implementation of the QZ algorithm applied to tropically scaled Lagrange linearization using well-separated tropical roots (8);

Alg.3 MATLAB's `polyeig` function;

Alg.4 `quadeig` from [10] when the degree of the matrix polynomial is 2;

Alg.5 Gaubert and Sharify's Algorithm [9, Alg. 1] (see also [18, Alg. 4.1]). We use the same MATLAB implementation as in [18].

We tested the five algorithms on all the square matrix polynomials from the NLEVP collection [5] having size $s \leq 300$. Table 2 only lists the NLEVP problems for which one of the algorithms returns a large backward error, that is, problems for which $\eta_P^{\max}(\textbf{Alg.j}) \geq 10ds\epsilon_{\text{mach}}$ for algorithm **Alg.j** as this indicates backward instability for this algorithm. The large backward errors are highlighted in bold in the table. For this set of test problems, all algorithms except `polyeig` (i.e., **Alg.3**) return eigenvalues with small backward errors. Recall that Gaubert and Sharify's algorithm (i.e., **Alg.5**) calls the QZ algorithm t times, where t is the number of tropical roots. Algorithms **Alg.1** and **Alg.2** only use one call to the QZ algorithm but our current implementation requires complex arithmetic even if the matrix polynomial has real coefficients. For all the problems in Table 2, the classical tropical roots are well-separated, which explains why the largest backward errors are the same for **Alg.1** and **Alg.2**.

We collected several matrix polynomials artificially constructed so as to have eigenvalues of widely varying magnitude. These test problems can be generated by the MATLAB function `get_P`[1]. Again we only display in Table 3 results for problems for which at least one of the algorithms

[1]The corresponding files can be downloaded from `https://people.cs.kuleuven.be/marc.vanbarel/downloads/get_P.zip`.

Table 2: Largest backward errors of eigenvalues computed by algorithms **Alg.1**–**Alg.5** on $s \times s$ matrix polynomials of degree d from the NLEVP collection. The last column lists the number t of tropical roots associated with the matrix polynomial.

Problem	d	s	**Alg.1**	**Alg.2**	**Alg.3**	**Alg.4**	**Alg.5**	t
`cd_player`	2	60	4.1e-16	4.1e-16	**3.1e-10**	2.5e-16	1.9e-13	2
`damped_beam`	2	200	4.8e-16	4.8e-16	**2.6e-11**	2.9e-16	9.1e-17	1
`hospital`	2	24	3.9e-15	3.9e-15	**2.9e-13**	1.3e-15	2.6e-15	1
`metal_strip`	2	9	3.0e-16	3.0e-16	**4.1e-14**	6.8e-16	3.7e-16	2
`orr_sommerfeld`	4	64	1.5e-15	1.5e-15	**9.1e-08**	—	4.5e-15	2
`planar_waveguide`	4	129	2.7e-15	2.7e-15	**4.7e-12**	—	1.6e-14	2
`power_plant`	2	8	1.3e-16	1.3e-16	**5.3e-12**	4.2e-18	1.5e-18	1

Table 3: Largest backward errors of eigenvalues computed by algorithms **Alg.1**–**Alg.5** on $s \times s$ matrix polynomials of degree d generated by `get_P`. The last two columns list the number t of tropical roots associated with the matrix polynomial and the number $\tilde{t}$ of well-separated tropical roots .

Problem	d	s	**Alg.1**	**Alg.2**	**Alg.3**	**Alg.4**	**Alg.5**	t	$\tilde{t}$
`Problem 1`	7	4	3.0e-15	3.0e-15	**2.5e-02**	—	2.0e-14	7	7
`Problem 2`	7	4	8.5e-16	8.5e-16	**1.2e-01**	—	**3.5e-12**	7	7
`Problem 4`	2	5	6.1e-16	6.1e-16	**5.8e-06**	**1.6e-13**	1.4e-15	2	2
`Problem 5`	2	5	3.4e-16	3.4e-16	1.1e-16	**4.2e-12**	1.5e-16	2	2
`Problem 7`	2	10	3.2e-16	3.2e-16	2.1e-16	**1.3e-02**	1.6e-16	2	2
`Problem 8`	2	10	2.4e-16	2.4e-16	6.4e-15	**5.4e-12**	1.5e-16	2	2
`Problem 9`	2	40	5.6e-16	5.6e-16	**7.6e-07**	1.3e-15	2.5e-16	2	2
`Problem 10`	5	20	8.8e-16	8.8e-16	**5.4e-12**	—	2.6e-15	2	2
`Problem 11`	10	8	2.4e-15	1.5e-15	**2.9e-12**	—	**1.8e-12**	4	3
`Problem 12`	4	30	1.1e-15	1.1e-15	**1.5e-12**	—	1.5e-15	2	2
`Problem 13`	4	9	1.1e-14	4.0e-15	**4.6e-12**	—	1.5e-15	3	2
`Problem 17`	10	2	2.9e-15	2.9e-15	**1.2e-01**	—	**7.7e-14**	10	10
`Problem 18`	4	4	9.5e-16	1.1e-15	**5.7e-11**	—	4.1e-15	2	2
`Problem 19`	4	4	6.0e-16	6.0e-16	**9.8e-14**	—	3.1e-16	2	2
`Problem 20`	5	4	1.2e-15	1.2e-15	**3.4e-06**	—	**1.4e-11**	3	3
`Problem 21`	5	4	7.5e-16	7.5e-16	**2.7e-07**	—	**4.7e-10**	3	3
`Problem 22`	4	4	1.4e-15	1.4e-15	**6.5e-07**	—	**8.2e-13**	3	3

returns a backward error larger than $10ds\epsilon_{\text{mach}}$. We also exclude scalar polynomials ($s = 1$) and `Problem 14`, which is the `orr_sommerfeld` problem from NLEVP (see Table 2). Such large backward errors are highlighted in bold. The quadratic eigensolver `polyeig` with the default parameters guarantees to return eigenvalues with small backward errors when $t = 1$, which is not the case for these problems. Gaubert and Sharify's algorithm (**Alg.5**) computes eigenvalues with large backward errors for some of these problems whereas the backward errors are small for the eigenvalues computed by both **Alg.1** and **Alg.2**. Note that $t > \tilde{t}$ for `Problem 11` and `Problem 13` and for both problems, $\eta_P^{\max}(\textbf{Alg.1}) > \eta_P^{\max}(\textbf{Alg.2})$. We show in the next set of experiments that $\eta_P^{\max}(\textbf{Alg.1}) \gg \eta_P^{\max}(\textbf{Alg.2})$ can happen.

Large backward errors tend to occur when the nonzero blocks in the linearization have widely varying magnitude (see [12, 14, 21]). We showed in Section 4 that for the matrices $\beta_i P(\sigma_i)/\sigma_i$ in the first block row of the Lagrange linearization to have norm one, the tropical roots need to be well-separated. We also showed numerically in Section 4 that when the tropical roots are close to each other, the matrices $\beta_i P(\sigma_i)/\sigma_i$ can have large norms. This can have an impact on the backward error as we now illustrate using Example 4 from [9]. These matrix polynomials can be generated using the MATLAB commands

```
s = 8; d = 10; scaling = [-5,-2,-3,-4,2,0,3,-3,4,2,5];
for i = 1:d+1
    P{i} = randn(s) * 10^scaling(i);
end
```

We generated 100 samples and computed their eigenvalues with algorithm **Alg.1** which uses classical tropical roots and algorithm **Alg.2** which uses well-separated tropical roots. The largest back-

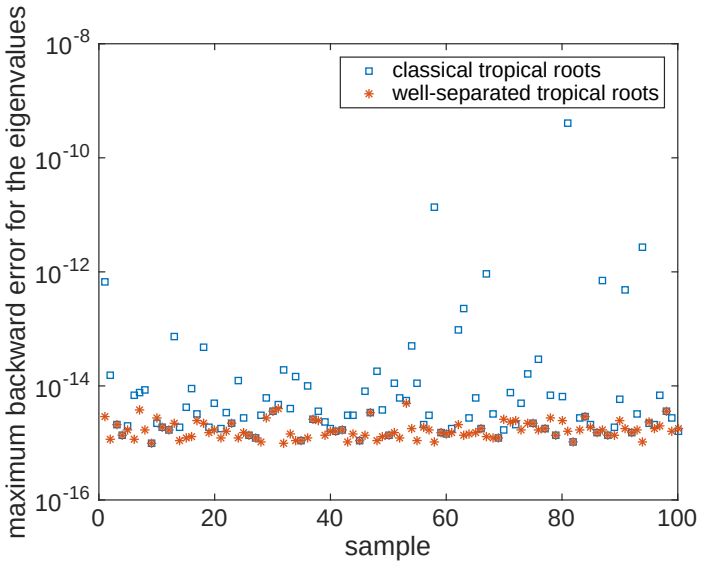

Figure 7: $\eta_P^{\max}$(**Alg.1**) (blue squares) versus $\eta_P^{\max}$(**Alg.2**) (red stars) for 100 samples of Example 4 from [9].

ward errors for each sample are plotted in Figure 7. The plot shows that for backward stability, it is important to use well-separated tropical roots when constructing the Lagrange linearization. Hence algorithm **Alg.2** is preferred to algorithm **Alg.1**.

7 Concluding remarks

We provided strong numerical evidence that indicates that for matrix pencils $A - zB$ for which A has entries of magnitude one and B is block diagonal with diagonal blocks allowed to have large variation in the magnitude of their norms (i.e., $\|B_{ii}\| \ll \|B_{jj}\|$ or $\|B_{ii}\| \gg \|B_{jj}\|$, $i \neq j$), the QZ algorithm with an appropriate stopping criterion computes all the finite and well-conditioned eigenvalues to high relative accuracy even when these eigenvalues are very different in magnitude. We showed that for computing eigenvalues of a matrix polynomial $P(z)$, a Lagrange linearization $A - zB$ of P can be built with A and B having the above properties. For this we introduced the notion of well-separated tropical roots and used them as interpolation nodes in the Lagrange linearization (3). We showed that this choice gives an upper bound for the norms of the blocks in the top block row of order one leading to a matrix pencil $A - zB$ with the specific desired structure. We performed numerical experiments showing that our eigensolver for polynomial eigenvalue problems computes eigenvalues with backward errors close to the machine precision even when these eigenvalues are very different in magnitude, a property not shared by existing methods based on linearization.

Several of the results of this manuscript are only evidenced by numerical experiments. In future work, we will try to define in a rigorous way the structure of the matrix pencil allowing the QZ algorithm to compute all (well-conditioned) eigenvalues with a high *relative* precision. As we have demonstrated numerically in this paper, this is connected to a graded structure in the backward error generated by the QZ algorithm. Currently a lot of research is done proving the backward stability of applying the QZ algorithm to certain types of linearizations, e.g., [8]. However, because only the well-known backward error properties of the QZ algorithm are used, the backward error on the matrix polynomial is one where all the coefficients are taken together normwise, without taking into account the possible difference in norm of each of the coefficients separately. This implies that when the norms of these coefficients are very different in magnitude, only part of the eigenvalues can be computed accurately and a different scaling with a new application of the QZ algorithm is necessary to compute another part of the spectrum. Taking into account the graded structure of the backward error when applying the QZ algorithm to certain structured matrix pencils, it could be possible to obtain a backward relative error of the size of the machine precision

on each of the coefficients of the matrix polynomial separately.

The number of different well-separated tropical roots is always equal to or smaller than the number of classical tropical roots. Hence, when using tropical scaling together with the block companion linearization, the number of different applications of the QZ algorithm can be reduced if the number of well-separated tropical roots is smaller. Note that this type of scaling can also be used even when solving the scalar polynomial problem when the classical balancing can not be applied on the companion matrix, e.g., when this balancing destroys the structure allowing for a fast implementation of the QR algorithm based on the structure of the companion matrix. Very recently a fast implementation of the QZ algorithm for the block companion pencil was developed in [3]. A fast scalar implementation of the QR algorithm for the companion matrix can be found in [4]. Both algorithms are proven to be *normwise* stable. However, as this manuscript shows, when the eigenvalues differ a lot in magnitude, the goal should be a fast algorithm having *componentwise* backward stability.

References

[1] Advanpix. Multiprecision Computing Toolbox for MATLAB. `http://www.advanpix.com`.

[2] A. Amiraslani, R. M. Corless, and P. Lancaster. Linearization of matrix polynomials expressed in polynomial bases. *IMA J. Numer. Anal.*, 29:141–157, 2009.

[3] J. L. Aurentz, T. Mach, L. Robol, R. Vandebril, and D. S. Watkins. Fast and backward stable computation of the eigenvalues of matrix polynomials. https://arxiv.org/abs/1611.10142, 2016.

[4] J. L. Aurentz, T. Mach, R. Vandebril, and D. S. Watkins. Fast and backward stable computation of roots of polynomials. *SIAM J. Matrix Anal. Appl.*, 36(3):942–973, 2015.

[5] T. Betcke, N. J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Software*, 39(2):7:1–7:28, Feb. 2013.

[6] D. A. Bini, V. Noferini, and M. Sharify. Locating the eigenvalues of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 34(4):1708–1727, 2013.

[7] R. M. Corless. Generalized companion matrices in the Lagrange basis. In L. Gonzalez-Vega and T. Recio, editors, *Proceedings EACA*, pages 317–322. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, June 2004.

[8] F. M. Dopico, P. Lawrence, J. Pérez, and P. Van Dooren. Block Kronecker linearizations of matrix polynomials and their backward errors. MIMS EPrint 2016.34, The University of Manchester, 2016.

[9] S. Gaubert and M. Sharify. Tropical scaling of polynomial matrices. In *Positive systems*, volume 389 of *Lecture Notes in Control and Information Sciences*, pages 291–303. Springer-Verlag, Berlin, 2009.

[10] S. Hammarling, C. J. Munro, and F. Tisseur. An algorithm for the complete solution of quadratic eigenvalue problems. *ACM Trans. Math. Software*, 39(3):18:1–18:19, Apr. 2013.

[11] D. J. Higham and N. J. Higham. Structured backward error and condition of generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 20(2):493–512, 1998.

[12] N. J. Higham, R.-C. Li, and F. Tisseur. Backward error of polynomial eigenproblems solved by linearization. *SIAM J. Matrix Anal. Appl.*, 29(4):1218–1241, 2007.

[13] N. J. Higham, D. S. Mackey, and F. Tisseur. The conditioning of linearizations of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 28(4):1005–1028, 2006.

[14] N. J. Higham, D. S. Mackey, F. Tisseur, and S. D. Garvey. Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems. *Internat. J. Numer. Methods Eng.*, 73(3):344–360, 2008.

[15] P. Lancaster and P. Psarrakos. A note on weak and strong linearizations of regular matrix polynomials. MIMS EPrint 2006.72, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, May 2006.

[16] P. Lawrence, M. Van Barel, and P. Van Dooren. Backward error analysis of polynomial eigenvalue problems solved by linearization. *SIAM J. Matrix Anal. Appl.*, 37(1):123–144, 2016.

[17] D. S. Mackey, N. Mackey, and F. Tisseur. Polynomial eigenvalue problems: Theory, computation, and structure. In P. Benner, M. Bollhöfer, D. Kressner, C. Mehl, and T. Stykel, editors, *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, pages 319–348. Cham Heidelberg New York Dordrecht London, 2015.

[18] V. Noferini, M. Sharify, and F. Tisseur. Tropical roots as approximations to eigenvalues of matrix polynomials. *SIAM J. Matrix Anal. Appl.*, 36(1):138–157, 2015.

[19] M. Sharify. *Scaling Algorithms and Tropical Methods in Numerical Matrix Analysis: Application to the Optimal Assignment Problem and to the Accurate Computation of Eigenvalues.* PhD thesis, Ecole Polytechnique, Palaiseau, France, Sept. 2011.

[20] M. Sharify and F. Tisseur. Effect of tropical scaling on linearizations of matrix polynomials: backward error and conditioning. MIMS EPrint, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2013. In preparation.

[21] F. Tisseur. Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl.*, 309(1-3):339–361, 2000.

[22] R. Van Beeumen, W. Michiels, and K. Meerbergen. Linearization of lagrange and hermite interpolating matrix polynomials. *IMA J. Numer. Anal.*, 2014.